

# **MANUEL INSTALLATION**

## **de**

# **supLeger et subLegerWebJSF**

## **V1.0**

### **Copyright et avertissement.**

Ces spécifications et l'implémentation du superviseur léger sont libres de tout droit.

L'utilisateur utilise ce produit à ses risques. La licence qui s'applique est la licence LGPL.

L'auteur décline sa responsabilité sur un éventuel mauvais fonctionnement du produit et des effets induits.

## Suivi des modifications

<i>Auteur</i>	<i>Version</i>	<i>Date</i>	<i>Modifications</i>
JL Pasturel	V1.0	30/03/2006	Initialisation

## Table des matières

1 BUT DE L'OUTIL.....	4
1.1 Remarque préliminaire.....	4
1.2 Définition rapide de l'outil.....	4
1.2.1 Partie supLeger.....	4
1.2.2 Partie supLegerWebJSF.....	4
1.3 Architecture et principes de fonctionnement.....	4
1.3.1 Partie supLeger.....	4
1.3.2 Partie supLegerWebJSF.....	5
2 MANUEL INSTALLATIONS .....	8
2.1 INTRODUCTION.....	8
2.2 INSTALLATION supLeger coté serveur.....	8
2.2.1 Pré-requis.....	8
2.2.2 Installation Postgres .....	8
2.2.3 Installation de Hibernate 2.1.....	8
2.2.4 Installation de la partie Serveur de supLeger.....	9
2.2.5 INSTALLATION supLeger coté agent.....	9
2.2.6 Installation de MSYS pour Windows ( Optionnel).....	11
2.3 INSTALLATION de la partie supLegerWebJSF.....	11
3 VERIFICATION DES INSTALLATIONS ET LANCEMENTS.....	12
3.1 Vérification bon fonctionnement de la base postgresQL.....	12
3.2 Création de la base .....	12
3.3 Lancement du serveur de collecte des données.....	12
3.4 Lancement de l'agent sur les serveurs à superviser .....	12
3.5 Lancement supLegerWebJSF.....	13
4 UTILISATION AVANCEE.....	16
4.1 Architecture séparée de la database, de supLeger et supLegerJSF.....	16
4.1.1 Schéma de l'architecture envisagée .....	16
4.1.2 Paramétrage de supLegerWebJSF.....	18
4.2 Internationalisation.....	19
4.3 Utilisation d'autres WAS.....	20
4.4 Utilisation d'une version antérieure de Postgres ( 7.4 par exemple).....	20
4.5 Joindre l'auteur.....	20
5 ANNEXE : Licence LGPL.....	21

# 1 BUT DE L'OUTIL

## 1.1 Remarque préliminaire

Le but essentiel lors de la réalisation de cet outil, était pour moi de prendre des compétences autour des Java Server Faces. D'autre part, j'avais déjà réalisé cet outil avec echo/echopoint V1( que je n'ai pas publié), j'ai donc repris la partie accès et gestions des données dans la database avec Hibernate V2.1. L'architecture logicielle peut effectivement prêter à discussion.

## 1.2 Définition rapide de l'outil

L'outil se compose de deux parties :

### 1.2.1 Partie supLeger

Il s'agit de réaliser un superviseur de fonctionnement de machine UNIX ou Windows composé d'un agent fonctionnant sur des serveurs et d'un serveur collationnant les informations de tous les serveurs.

### 1.2.2 Partie supLegerWebJSF

Cette partie est chargée d'exploiter ces données en réalisant les fonctions de superviseur quasi temps réel et superviseur temps différé

**Notas :**

- Dans ce document nous proposerons une installation rapide pour les gens habitués aux installations
- Licence du produit : LGPL voir en annexe la le contenu de la licence.

## 1.3 Architecture et principes de fonctionnement

Le **schéma 1**, à la fin de ce paragraphe, donne l'architecture logicielle complète des deux parties de ce superviseur léger.

### 1.3.1 Partie supLeger

La partie supLeger, est chargé de la collecte des données indicateurs des applications et des systèmes implantés sur des serveurs. L'agent et serveur de supervision sont écrits en **Java**. Il faut une **JVM 1.3** au moins coté **agent** et une **JVM 1.4** au moins coté **superviseur**.

1 serveur peut accueillir n applications. On peut définir p indicateurs par application. L'agent permet de recoller ces indicateurs. Le collecteur de **supLeger** peut accueillir les données de supervisions de plusieurs serveurs.

#### • Côté Agent

La productions de ces indicateurs peut se faire de 2 façons sur chaque serveur et application :

- cas 1 : écriture mode **SHELL** : en shell Unix, batch Windows, programme C, java, perl, VB ...
- cas 2 : écriture de classe **JAVA** appelée directement depuis l'agent.( Utilisation du mécanisme de réflexion Java)

Dans les deux cas, il faut que le retour soit de la forme :

**<retour\_forme\_string>;<données\_complémentaires\_forme\_string>**

Le `retour_forme_string` est toujours un String et peut contenir des données de type int, float, double, Date ( avec un certain format), String ; elles seront interprétées en fonction d'un fichier de configuration qui est défini dans le manuel utilisateur..

La partie données complémentaires permet de retour de données contextuelles à la requête qui complète le résultat.

Dans le cas 1, le retour doit être fait sur le stdout ( fonction **echo** pour les shells unix et batch windows , fonction **printf** pour les programmes C, fonction **System.out.println()** pour les programmes java ...par exemple).

Dans le cas 2, la classe java, possède une fonction «**retour**» qui retourne un string au format vue plus haut . Voir dans le manuel utilisateur pour la création de telle fonction;

On aura compris qu'il faut qu'une seule écriture dans le **stdout** !

A partir de ces retours l'**agent** constitue des messages qu'il envoie au **superviseur** central.

### • Côté Superviseur

Le superviseur est chargé de recoller les données des agents à travers une communication en mode TCP/IP sur socket et de les charger dans la base de données.

Pour cela il est utilisé, le logiciel de mapping **Hibernate**, qui à partir des objets métiers de granulométrie fine mappe ces objets dans la database. La database est très simple puisqu'elle comporte simplement **10 tables** ( en y incluant les tables de relations n:n). Le moteur de database choisie est **PostgreSQL V8** qui est disponible maintenant en natif sous windows ( plus besoin de cygwin)

Le nombre de POJOs est de **8**.

## 1.3.2Partie supLegerWebJSF

C'est la partie visible pour l'utilisateur. Elle permet de réaliser la visualisation des indicateurs soit en mode continue ( fonction de **supervision quasi temps réel**) ou en mode différé et sur une période paramétrable par l'utilisateur ( fonction **supervision temps différé**).

Cette partie exploite les données recollée par la partie **supLeger**.

Le moteur du Web Application Serveur utilisé est **Winstone** . Il s'agit d'un container light de servlet/JSP ( sous double licence LGPL et CDDL) que l'on peut trouver à cette adresse :

<http://winstone.sourceforge.net/>

Je dois remercier l'auteur ( **Rick Knowles** ) qui m'a beaucoup aidé pour intégrer mon outil dans son serveur ( qu'il n'avait à priori pas conçu pour accueillir les JSF, et Rick Knowles m'a permis de réaliser cette intégration).

Dans le paragraphe 4, j'indique comment intégrer mon outils dans d'autres contenue de Servlet/JSP basés sur Tomcat ( ex JONAS).

Pour la visualisation des données, on a utilisé un socle commun de framework :le framework JSF ( Java

Server Faces ) et son implémentation Apache MyFaces version 1.1.1

Ce framework est sous licence Apache V2.0 .

**MyFaces** : <http://myfaces.apache.org/>

Pour découpler la partie vue de la partie modèle, l'aspect business et manipulation des données est délégué aux objets **façades gros grains**. Pour ce projet, j'ai gardé des adhérences entre la partie vue et métiers pour assurer une meilleure performance ( je n'ai pas utilisé de pattern Data Transfert Object.). C'est à dire que pour utiliser les façades dans un autre contexte de visualisation ( Client lourd, , Struts ...), il faudrait retoucher légèrement au code des façades.

La gestion de l'accès aux données et persistance ( en lecture seule en fait) passe aussi par le mécanisme **POJOs / Hibernate et database PostgreSQL**.

Nota : ce sont les mêmes **pojos** que ceux de la partie **supLeger**, mais le code est répliqué car on peut avoir des architectures où le superviseur de **supLeger** n'est pas sur la même machine que le WAS (Winstone ) de **subLegerWeb**. Pour cette même raison le framework **Hibernate** est dupliqué dans **supLeger** et **supLegerWeb**.

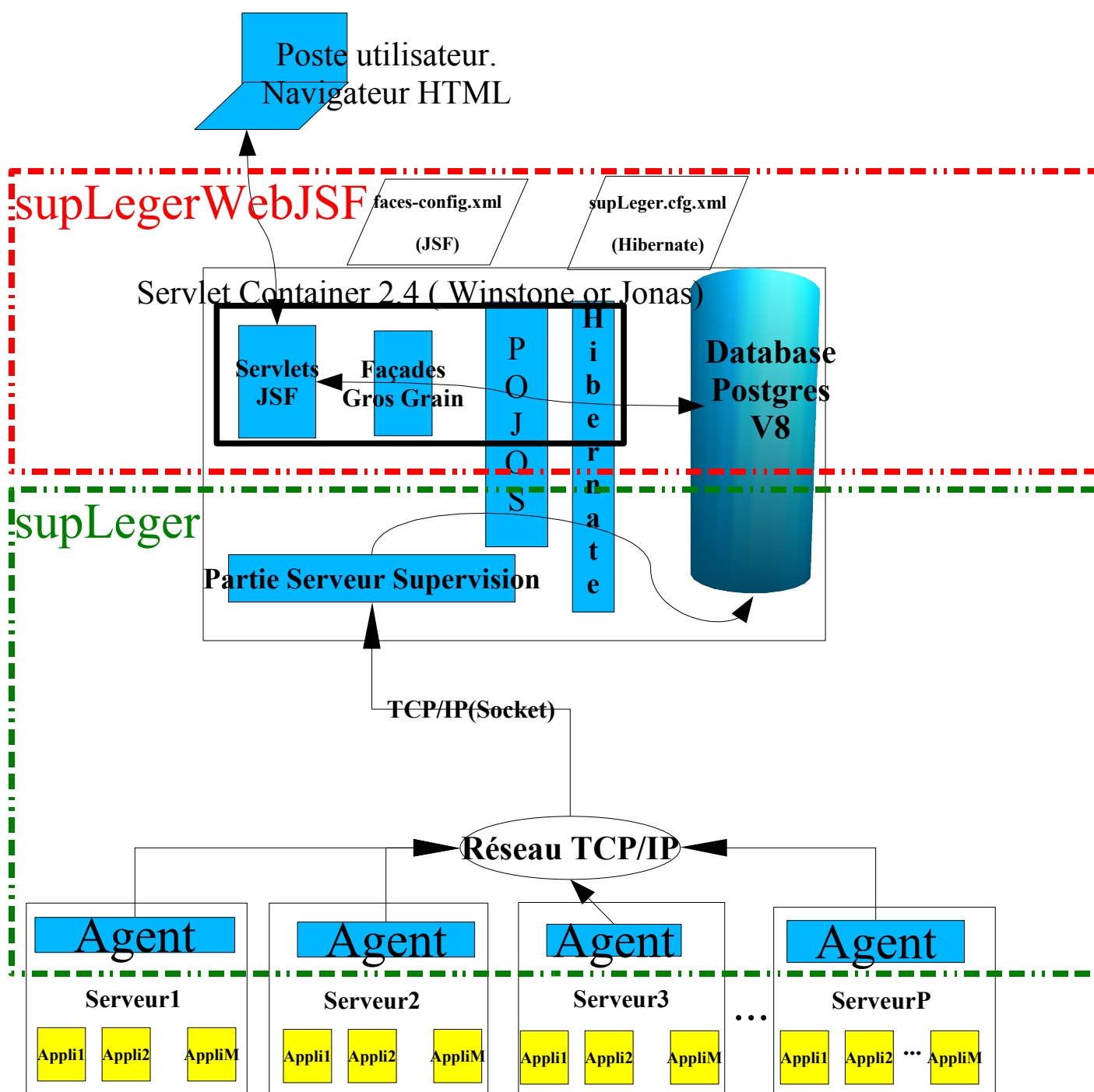


Schéma 1 : Architecture Logicielle

## 2 MANUEL INSTALLATIONS

### 2.1 INTRODUCTION

Nous allons décrire ci-dessous l'installation des composants sur une architecture Windows. Pour une installation sous \*nix, nous supposons que l'installateur sera capable d'adapter cette procédure.

Les logiciels utilisés proposent leur manuel d'installation propres pour tout type d'installation. Vous vous reporterez à la documentation fournie avec le logiciel pour plus de détails. Il ne sera indiqué dans ce manuel que les opérations essentielles pour l'installation.

Il y a deux d'installations à réaliser :

- installation de supLeger
- installation de supLegerWebJSF

### 2.2 INSTALLATION *supLeger coté serveur*

#### 2.2.1 Pré-requis

- PostgreSQL V8 pour Windows à télécharger de <http://www.postgres.org/>
- La distribution de supLeger soit **supLegerV100.jar**.
- Optionnel, pour faciliter l'écriture de shell sous Windows ( à l'aide de commande Unix-like) télécharger le package MSYS à partir de :<http://www.mingw.org/download.shtml>

Les exemples donnés de programmes sous Windows utilisent MSYS.

#### 2.2.2 Installation Postgres

- Installation de PostgreSQL ( Autoriser les connexions à distances, positionner le langage à votre langue (French) , et les codes caractères adaptés (LATIN1)
- Créer un utilisateur database de login **essai** et de password **essai** ( lui donner le droit de créer des databases)
- Créer un tablespace **tbs\_essai** sous l'arborescence **C:\PGDATA** ( propriétaire **essai**).
  - Sous Windows XP/NTFS on donne le droit d'accès à [C:\PGDATA](#) pour l'utilisateur windows **postgres** par la commande systeme sous compte administrateur ( à manier avec précaution, bien vérifier la syntaxe avant de lancer) :  
**cacls.exe /G / postgres:F /E**  
On peut ensuite créer le tablespace à l'aide de Pgadmin.
- Positionner le PATH pour atteindre les binaires de PostgreSQL ( chemin de la forme :  
**c:\Program Files\PostgreSQL\8.x\bin** )

#### 2.2.3 Installation de Hibernate 2.1

Les bibliothèques Hibernate seront amenées avec le package **supLegerV100.jar**. Si on veut faire une mise à jour on remplacera les librairies correspondantes de Hibernate avec celle de la dernière version .

## 2.2.4 Installation de la partie Serveur de supLeger

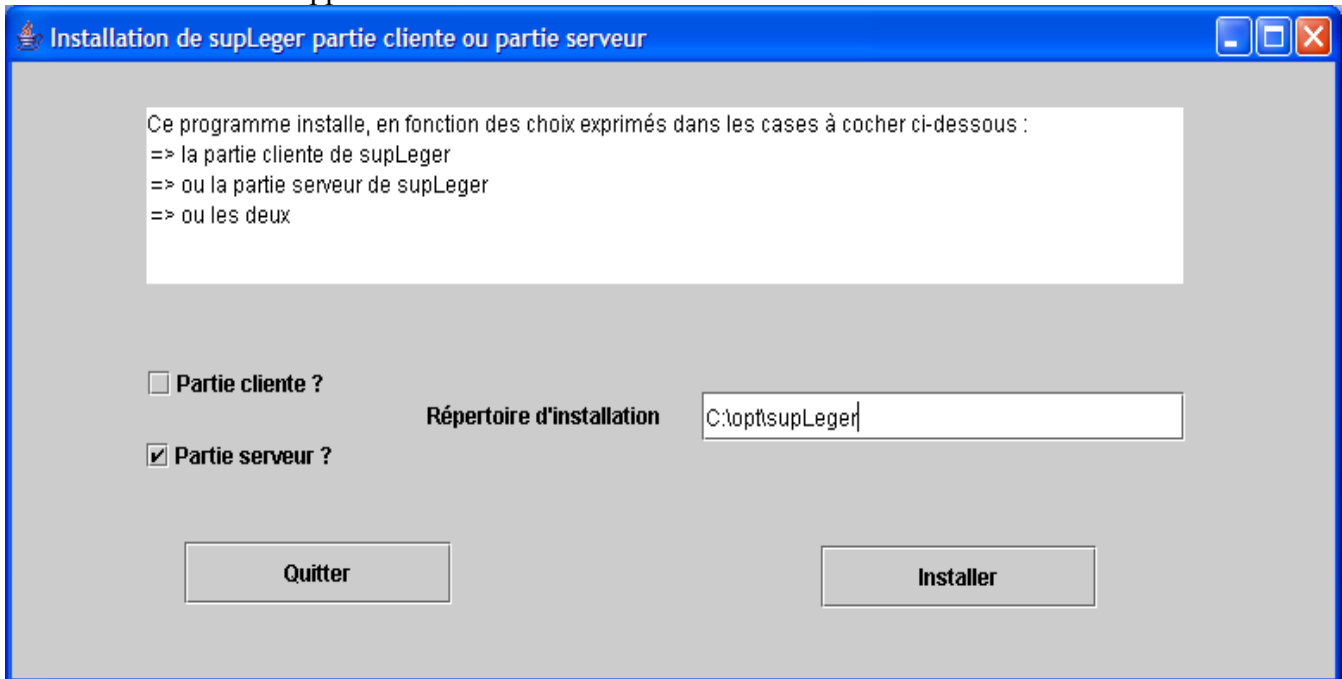
Créer le répertoire **c:\opt\supLeger**

Déposer **supLegerV100.jar** dans **c:\opt\supLeger**

Lancer les commandes pour Windows :

**java -classpath .\supLegerV100.jar Install**

L'écran ci-dessous apparaît :



Servir le champ du répertoire d'installation et cocher la checkbox de la partie serveur et cliquer sur **Installer**.

pour \*nix :

**java -classpath ./supLegerV100.jar Install**

A la fin de l'installation on peut supprimer supLegerV100.jar.

## 2.2.5 INSTALLATION supLeger coté agent

**Pré-requis :** Disposer de la distribution de supLeger soit **supLegerV100.jar**.

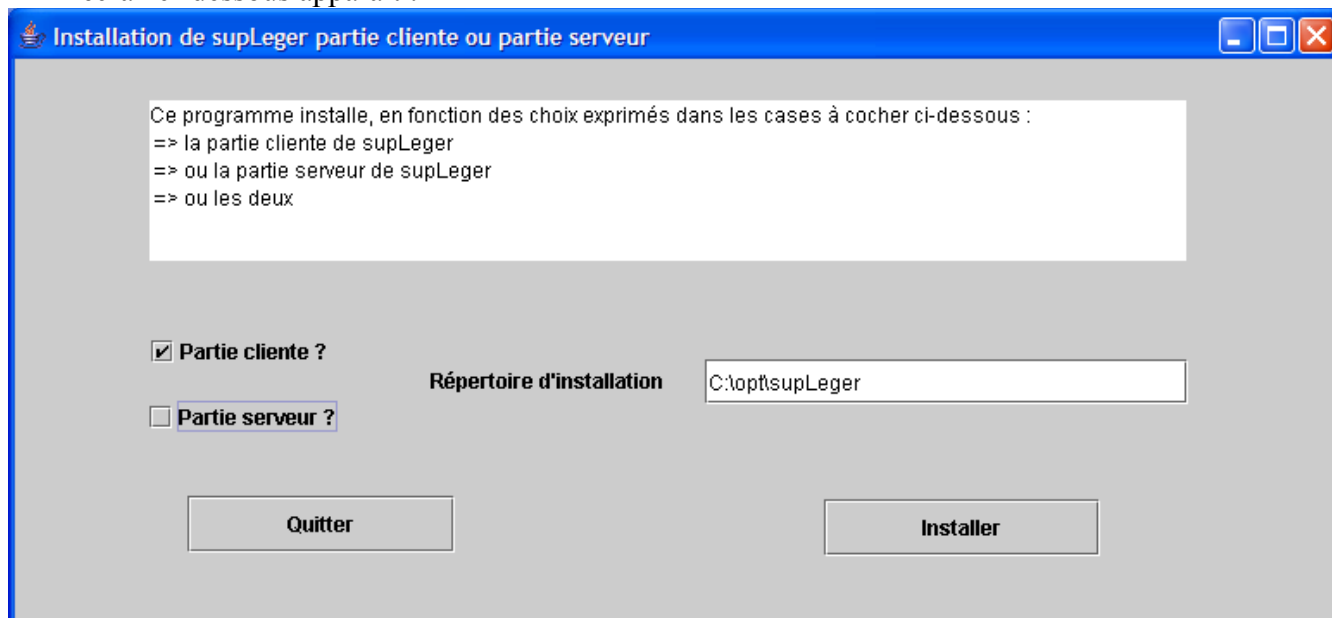
Créer le répertoire **c:\opt\supLeger\** ou **/opt/supLeger**

Déposer **supLegerV100.jar** dans **c:\opt\supLeger** (ou **/opt/supLeger**)

Lancer les commandes pour **Windows** :

**java -classpath .\supLegerV100.jar Install**

L'écran ci-dessous apparaît :



Servir le champ du répertoire d'installation et cocher la checkbox de la partie cliente et cliquer sur **Installer**.

pour **\*nix** :

**java -classpath ./supLegerV100.jar Install**

A la fin de l'installation on peut supprimer supLegerV100.jar.

**Nota :** On pourra installer aussi un agent sur la machine du serveur, pour tester le bon fonctionnement de l'ensemble et surveiller par la suite l'évolution de la database Postgres en fonctionnement normal.

### 2.2.6 Installation de MSYS pour Windows ( Optionnel)

Le package MSYS contient le portage des principales commandes Unix, qui peuvent être exécutées depuis des fichiers .bat de Windows.

Site de téléchargement : <http://www.mingw.org/download.shtml>

L'archive récupérée est :

**MSYS-<version>..exe**

L'installer sous [c:\msys](#)

Il existe aussi des utilitaires **pstools** qui donnent aussi des commandes de contrôles des états des processus et de la mémoires sur le site :

<http://www.sysinternals.com/>

## 2.3 *INSTALLATION de la partie supLegerWebJSF*

On suppose que l'on installe la partie web application sur le meme serveur que la partie serveur de supLeger évoquée plus haut.

La partie **supLegerWebJSF** nécessite d'avoir la seule archive suivante :

**supLegerWebJSFWinstone.zip**

Créer le répertoire d'accueil de l'application web-application :

exemple : **c:\opt\winstone** sous windows ou **/opt/winstone** sous unix :

Dézipper le fichier **supLegerWebJSFWinstone.zip** sous ce répertoire

**Noter** : l'outil **java jar** permet de dézipper :

**jar -xvf supLegerWebJSFWinstone.zip**

## 3 VERIFICATION DES INSTALLATIONS ET LANCEMENTS

### 3.1 Vérification bon fonctionnement de la base postgresQL

Vérifier avec Pgadmin que vous pouvez vous connecter à la base. J'ai noté sous windows XP des problèmes de démarrage du service.

En cas d'échec de la connexion, il faut arrêter et relancer le service.

Ces actions sont accessibles depuis les menus programmes Postgres de Windows.

### 3.2 Création de la base

Lancer une commande console Windows.

Se positionner sous le répertoire **c:\opt\supLeger\backupdatabase**

Lancer la commande de connexion à la base:

**psql -U postgres template1**

Servir le mot de passe.

Sous le prompt psql lancer :

**\i createEssai.sql**

La base se crée . Puis lancer le script de population de la base pour intégrer des exemples de requêtes que l'on pourra ensuite utiliser dans les agents de **supLeger**.

**\i populate.sql**

### 3.3 Lancement du serveur de collecte des données.

Depuis l'explorateur de fichier, ou en ouvrant une console Windows lancer :

**C:\opt\supLeger\super\lanceServeurThread.bat**

Pour unix :

Rendre exécutable le fichier **lanceServeurThread.ksh**

**chmod 755 /opt/supLeger/super/lanceServeurThread.ksh**

puis lancer

**/opt/supLeger/super/lanceServeurThread.ksh**

On peut lancer en background et/ou nohup.

Vérifier que la console n'affiche pas d'erreur.

### 3.4 Lancement de l'agent sur les serveurs à superviser

Depuis l'explorateur de fichier, ou en ouvrant une console Windows lancer :

**C:\opt\supLeger\agent\lanceAgent.bat**

Pour unix :

Rendre exécutable le fichier **lanceAgent.bat.ksh**

**chmod 755 /opt/supLeger/agent/lanceAgent.bat.ksh**

puis lancer

**/opt/supLeger/agent/lanceAgent.bat.ksh**

On peut lancer en background et/ou nohup.

Vérifier que la console n'affiche pas d'erreur.

Quand les requêtes paramétrées avec des timers se lancent, on peut vérifier dans les consoles ( coté agent ou coté serveur) que les données sont biens insérées en base. On peut le vérifier aussi avec l'outil postgres : **PgAdmin**.

### **3.5Lancement *supLegerWebJSF***

Avant de lancer on vérifiera que les classpath et les path contenus dans le fichier **lanceW.bat** ou **lanceL.ksh** et on les modifiera au besoin.

D'autre part il faudra rendre exécutable sous linux/unix le fichier **lanceL.ksh** par la commande :

**chmod 755 lanceL.ksh**

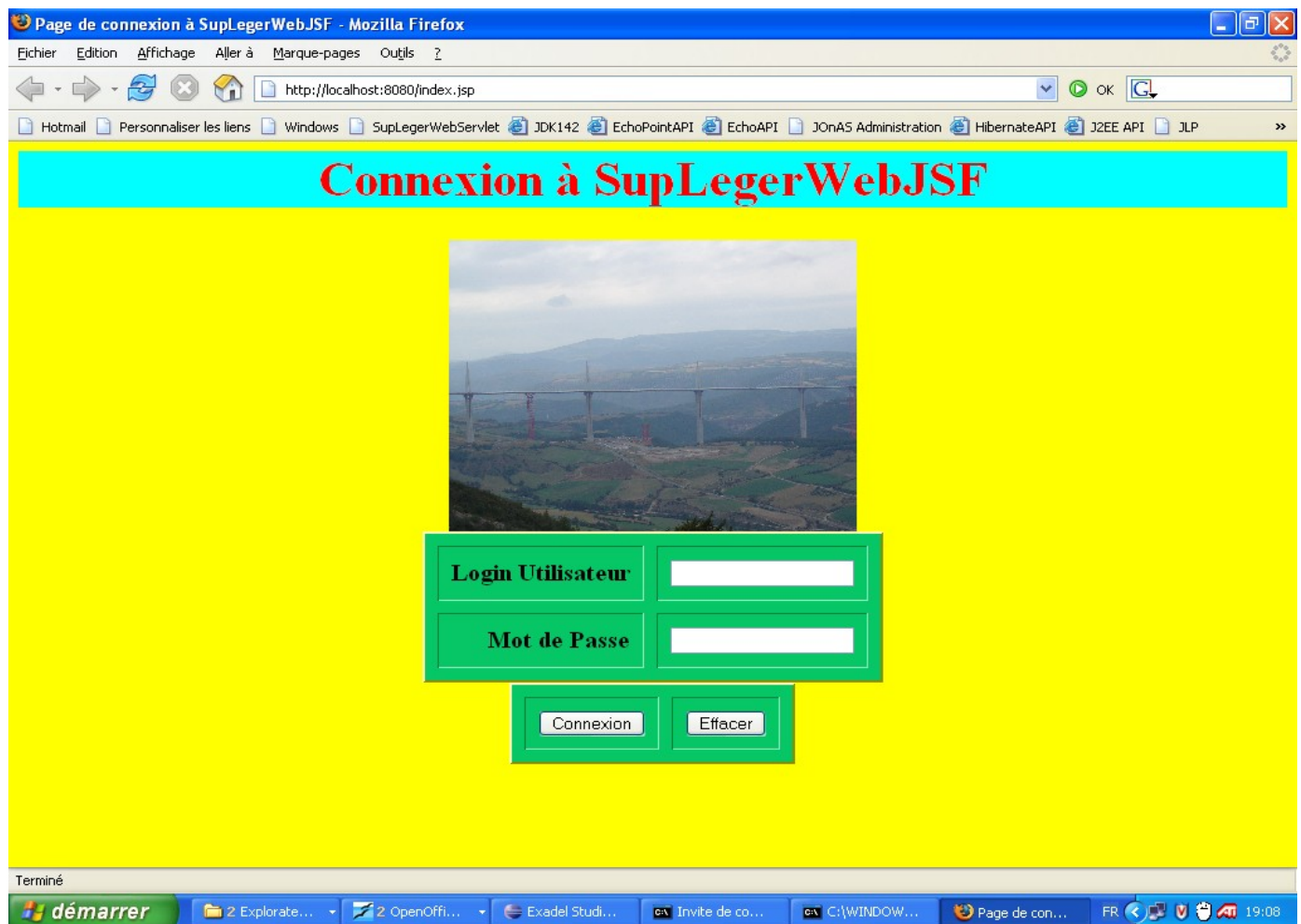
Il suffit de lancer sous le répertoire de **winstone** le fichier **lanceW.bat** ou **lanceW.ksh** en background.

Si le poste utilisateur abrite aussi le serveur **supLegerWebJSF** , on peut faire un test de connexion en lançant Internet Explorer ( ou FireFox ou Mozilla) et taper l'URL suivant :

<http://localhost:8080/>

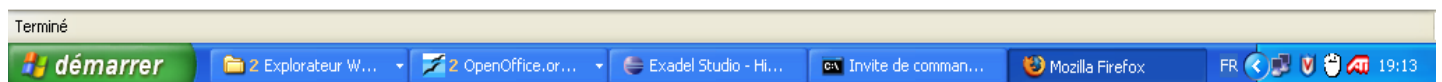
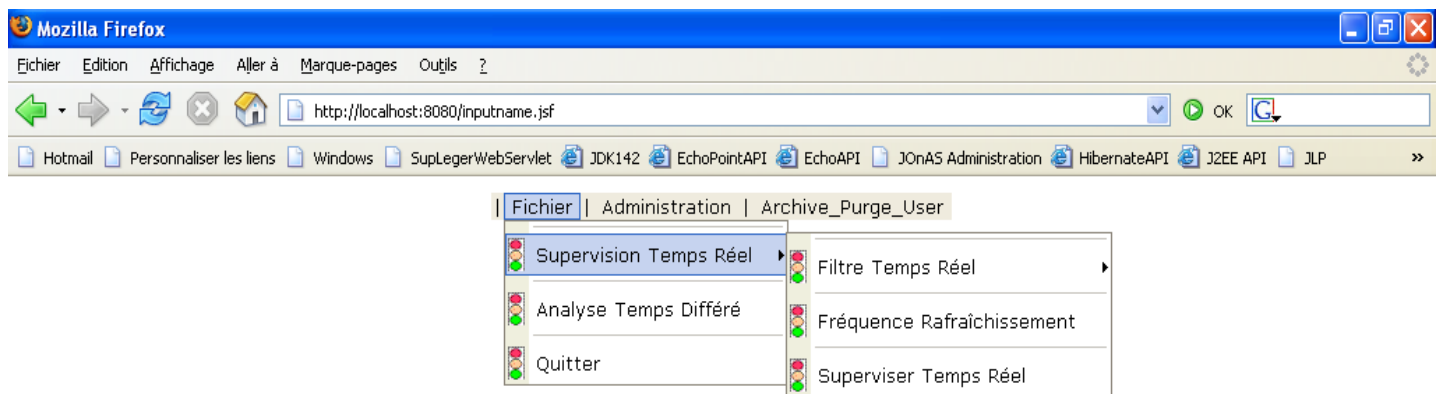
On peut aller jusqu'à la connexion administrateur, avec le login/password : **admin/admin**.

On peut enchaîner les écrans ci-dessous :



servir login/password => **admin/admin**

Pour information, la photo ci-dessus est le pont de Millau 6 mois avant son achèvement.  
cliquer sur **Connexion** et l'écran suivant du menu général apparaîtra :



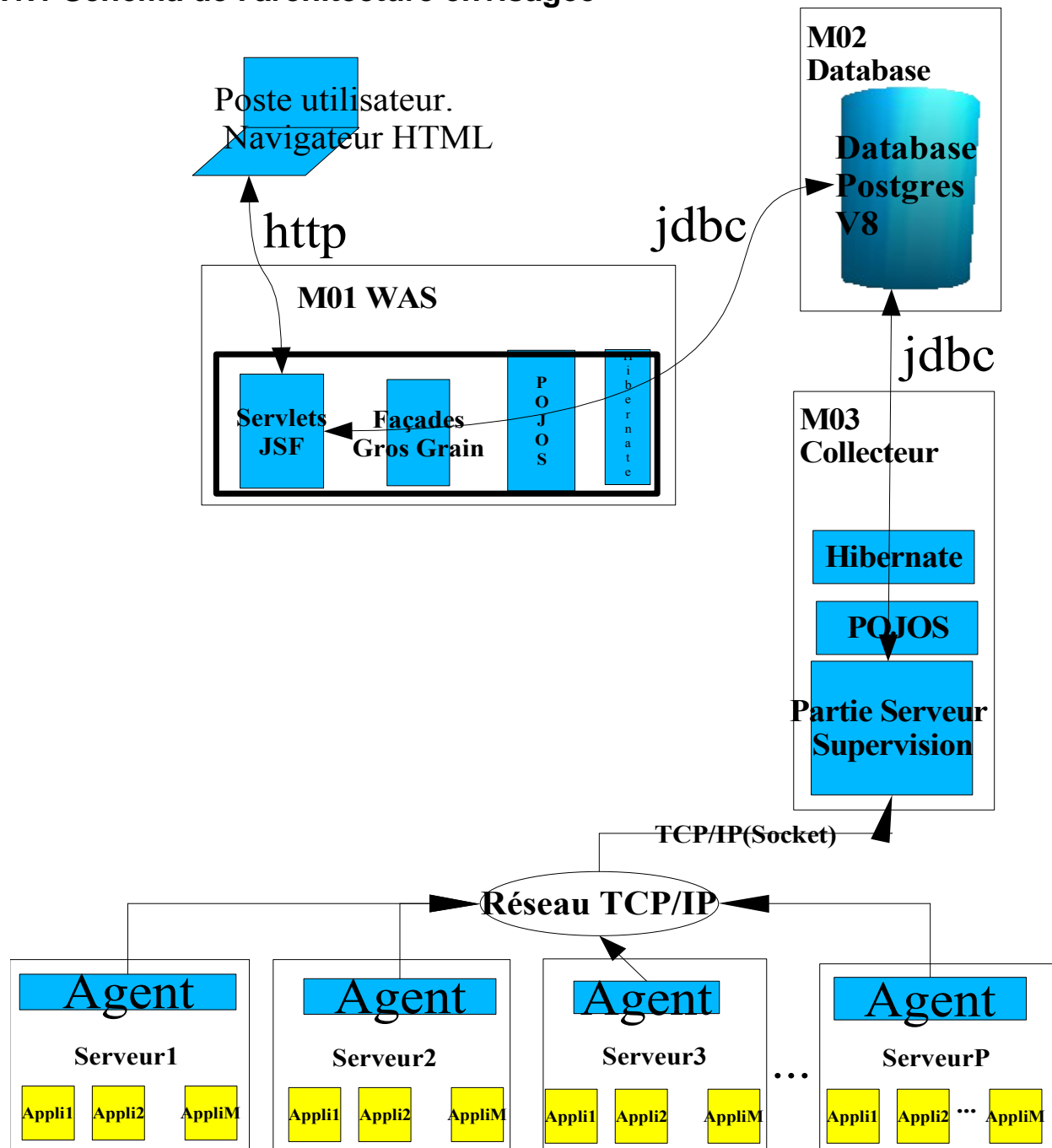
Si tous les tests, décrits ci-dessus sont passés, vos installations de **supLeger** et **supLegerWebJSF** sont correctes.

Pour la suite de l'utilisation, voir le manuel utilisateur.

## 4UTILISATION AVANCEE

### 4.1 Architecture séparée de la database, de supLeger et supLegerJSF

#### 4.1.1 Schéma de l'architecture envisagée



Cette architecture correspond au cas le plus complexe à 3 noeuds. Les autres cas intermédiaires pourront être facilement extrapolés à partir de cette architecture. Les paramètres ci-dessous correspondent à l'architecture ci-dessus.

Paramétrage supLeger ( serveur de collecte et agents ).

### Coté serveur de collecte :

Le fichier de Paramétrage est :

**C:\opt\supLeger\super\conf\supLeger.cfg.xml**

Le contenu est le suivant

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration
PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-2.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
<property name="hibernate.dialect">net.sf.hibernate.dialect.PostgreSQLDialect</property>
<property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/essai</property>
<property name="hibernate.connection.username">essai</property>
<property name="hibernate.connection.password">essai</property>
<property name="hibernate.c3p0.min_size">5</property>
<property name="hibernate.c3p0.max_size">20</property>
<property name="hibernate.c3p0.timeout">1800</property>
<property name="hibernate.c3p0.max_statement">50</property>

<property name="show_sql">>true</property>

<property name="use_outer_join">>false</property>

<!-- fichiers de mapping -->
<mapping resource="supLeger/superviseur/pojos/InstanceAppli.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/MessageArchive.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/MessageRecu.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Requete.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Role.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Serveur.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Utilisateur.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Derniermessage.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

La ligne en gras est à paramétrer, il faut donner l'adresse de la machine contenant la database en remplacement de **localhost** ( Une adresse de type DNS est préférable.)

## Coté Agent sur chaque serveur supervisé :

Ne pas oublier aussi sur chaque agent de donner l'adresse DNS du serveur de collecte le fichier de paramétrage est :

**C:\opt\supLeger\agent\conf\supLegerAgent.conf**

```
#IP_ADRESSE=127.0.0.1:5712
HOSTNAME=localhost
DNS_ADRESSE=localhost.localdomain:5712
SOCKET_LOCAL=5306
```

Il s'agit la aussi de remplacer l'adresse **localhost.localdomain** par l'adresse du serveur de collecte ( Donner une adresse DNS ou un alias accessible par le réseau)..

Le hostname correspond au nom du serveur, à changer pour identifier d'une manière unique le serveur ( il n'est pas besoin que cela soit le hostname réel du serveur).

### 4.1.2 Paramétrage de supLegerWebJSF

Il s'agit aussi de mettre à jour le fichier de configuration Hibernate qui se trouve dans l'archive **supLegerWebJSF.war**.

Pour cela il faut décompacter l'archive .war dans un répertoire à part.

Le décompactage peut se faire avec l'outil jar ou winzip ou unzip en fonction des Operating Systèmes.

Par exemple la commande :

**jar -xvf supLegerWebJSF.war**

décompacte le fichier. Supprimer le fichier **supLegerWebJSF.war** ( pour éviter de l'intégrer à la re-cr  ation de l'archive)

Il faut aller ensuite sous l'arborescence :

**<rep\_decompactage>supLegerWebJSF.war\WEB-INF\conf\**

ouvrir le fichier **supLeger.cfg.xml** ( qui est identique    celui vu plus haut) voir ci-dessous

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration
PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-2.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
<property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/essai</property>
<property name="hibernate.connection.username">essai</property>
<property name="hibernate.connection.password">essai</property>
<property name="hibernate.c3p0.min_size">5</property>
<property name="hibernate.c3p0.max_size">20</property>
<property name="hibernate.c3p0.timeout">1800</property>
```

```

<property name="hibernate.c3p0.max_statement">100</property>
<property name="hibernate.dialect">net.sf.hibernate.dialect.PostgreSQLDialect</property>
<property name="show_sql">false</property>
<property name="use_outer_join">true</property>

<!-- fichiers de mapping -->
<mapping resource="supLeger/superviseur/pojos/InstanceAppli.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/MessageArchive.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/MessageRecu.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Requete.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Role.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Serveur.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Utilisateur.hbm.xml"/>
<mapping resource="supLeger/superviseur/pojos/Derniermessage.hbm.xml"/>
</session-factory>
</hibernate-configuration>

```

et comme précédemment :

La ligne en gras est à paramétrer, il faut donner l'adresse de la machine contenant la database en remplacement de **localhost** ( Une adresse de type DNS est préférable.)

Sauver le fichier et recréer l'archive war par exemple par :

```
jar cvf c:\tmp\supLegerWebJSF.war .\*
```

récupérer le fichier war sous [c:\tmp](#) et repositionnez-le sous le répertoire adéquat de votre WAS. Par exemple sous **c:\opt\winstone** pour ma configuration.

## 4.2 Internationalisation

**SuplegerWebJSF** est livré dans cette version avec une interface Française et une interface Anglaise.. Le fichier au format Java properties est sous l'arborescence :

**\supLegerWebJSF\WEB-INF\classes\demo\bundle**

le fichier de configuration en langue anglaise est : **Messages\_en.properties**

A partir de ce fichier on peut créer les autres internationalisation ( les clés sont en français, mais à partir de l'anglais cela sera assez facile de créer les fichiers pour les autres langues). J'invite les personnes de langue anglaise à corriger certaines fautes que j'aurais pu commettre.

Ci-dessous un extrait de ce fichier :

```

inputname_header= Connect at SupLegerWebJSF
prompt=User Login
password=User Password
greeting_text=Hello
errorPassword= The length of Password must be greater than 6 caracters
connexion= Connection

```

```
fichier = File
administration = Administration
archivePurgeUser = Archive_Purge_User
supTempsReel = Supervision Real Time
supTempsDiff = Analyze Differed Time
quitter = Quit
edition = Edit
purgeGenerale = General Purge
archiveGenerale = General Archive
```

Il suffit ensuite de recréer l'archive comme indiqué au paragraphe précédent et de la replacer dans l'espace du WAS.

### 4.3 Utilisation d'autres WAS

Le WAS léger utilisé Winstone peut être remplacé par d'autres WAS. J'ai fait le test avec JONAS 4.6 et cela marche correctement il suffit de faire les 2 opérations suivantes :

- déposer le fichier **supLegerWebJSF.war** sous le répertoire **<JONAS\_BASE>/webapps/autoload**
- de déposer le driver JDBC ( **postgresql-8.1dev-403.jdbc3.jar** dans mon cas) de Postgres sous le répertoire **<JONAS\_BASE>/lib/ext**

Ce qui veut dire que cela doit marcher avec tous les WAS à base de Tomcat 5 ( Servlet 2.4 / JSP 2.0) devrait aussi faire l'affaire. Il n'y a pas d'utilisation d'EJB.

### 4.4 Utilisation d'une version antérieure de Postgres ( 7.4 par exemple)

Je fournis deux scripts de créations de base 7.4, situés sous **<server\_root>/super/backupdatase**, à lancer dans l'ordre sous compte DBA postgres, connecté à template1 :

- **createEssai74.sql**
- **populate74.sql**

Sur quelques tests réalisés, il n'est pas à priori nécessaire de changer de version de de driver JDBC, celui de la version 8 semble convenir.

### 4.5 Joindre l'auteur

Pour tout problème, ou tout bug vous pouvez me joindre à l'adresse e-mail :

[jean-louis.pasturel@wanadoo.fr](mailto:jean-louis.pasturel@wanadoo.fr)

## 5ANNEXE : Licence LGPL

### GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2,  
hence  
the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General  
Public  
Licenses are intended to guarantee your freedom to share and  
change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who decide to use it.  
You  
can use it too, but we suggest you first think carefully about  
whether  
this license or the ordinary General Public License is the better  
strategy to use in any particular case, based on the explanations  
below.

When we speak of free software, we are referring to freedom of  
use,  
not price. Our General Public Licenses are designed to make sure  
that  
you have the freedom to distribute copies of free software (and  
charge  
for this service if you wish); that you receive source code or can  
get  
it if you want it; that you can change the software and use pieces of  
it in new free programs; and that you are informed that you can do  
these things.

To protect your rights, we need to make restrictions that forbid  
distributors to deny you these rights or to ask you to surrender  
these  
rights. These restrictions translate to certain responsibilities for  
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis  
or for a fee, you must give the recipients all the rights that we gave  
you. You must make sure that they, too, receive or can get the

source

code. If you link other code with the library, you must provide  
complete object files to the recipients, so that they can relink them  
with the library after making changes to the library and  
recompiling  
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright  
the  
library, and (2) we offer you this license, which gives you legal  
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that  
there is no warranty for the free library. Also, if the library is  
modified by someone else and passed on, the recipients should  
know  
that what they have is not the original version, so that the original  
author's reputation will not be affected by problems that might be  
introduced by others.

Finally, software patents pose a constant threat to the existence of  
any free program. We wish to make sure that a company cannot  
effectively restrict the users of a free program by obtaining a  
restrictive license from a patent holder. Therefore, we insist that  
any patent license obtained for a version of the library must be  
consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the  
ordinary GNU General Public License. This license, the GNU  
Lesser  
General Public License, applies to certain designated libraries, and  
is quite different from the ordinary General Public License. We  
use  
this license for certain libraries in order to permit linking those  
libraries into non-free programs.

When a program is linked with a library, whether statically or  
using  
a shared library, the combination of the two is legally speaking a  
combined work, a derivative of the original library. The ordinary  
General Public License therefore permits such linking only if the  
entire combination fits its criteria of freedom. The Lesser General  
Public License permits more lax criteria for linking other code  
with  
the library.

We call this license the "Lesser" General Public License because  
it  
does Less to protect the user's freedom than the ordinary General  
Public License. It also provides other free software developers  
Less  
of an advantage over competing non-free programs. These  
disadvantages

are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

#### GNU LESSER GENERAL PUBLIC LICENSE

##### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from

such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License,

version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative

work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses

terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is

implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT

PERMITTED BY APPLICABLE LAW.  
EXCEPT WHEN OTHERWISE STATED IN WRITING THE  
COPYRIGHT HOLDERS AND/OR  
OTHER PARTIES PROVIDE THE LIBRARY "AS IS"  
WITHOUT WARRANTY OF ANY  
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING,  
BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND  
FITNESS FOR A PARTICULAR  
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND  
PERFORMANCE OF THE  
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE  
DEFECTIVE, YOU ASSUME  
THE COST OF ALL NECESSARY SERVICING, REPAIR OR  
CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE  
LAW OR AGREED TO IN  
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY  
OTHER PARTY WHO MAY MODIFY  
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED  
ABOVE, BE LIABLE TO YOU  
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL,  
INCIDENTAL OR  
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE  
OR INABILITY TO USE THE  
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF  
DATA OR DATA BEING  
RENDERED INACCURATE OR LOSSES SUSTAINED BY  
YOU OR THIRD PARTIES OR A  
FAILURE OF THE LIBRARY TO OPERATE WITH ANY  
OTHER SOFTWARE), EVEN IF  
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED  
OF THE POSSIBILITY OF SUCH  
DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest  
possible use to the public, we recommend making it free software  
that  
everyone can redistribute and change. You can do so by  
permitting  
redistribution under these terms (or, alternatively, under the terms  
of the  
ordinary General Public License).

To apply these terms, attach the following notices to the library.  
It is  
safest to attach them to the start of each source file to most  
effectively  
convey the exclusion of warranty; and each file should have at  
least the  
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it  
does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or  
modify it under the terms of the GNU Lesser General Public  
License as published by the Free Software Foundation; either  
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied  
warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR  
PURPOSE. See the GNU  
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General  
Public  
License along with this library; if not, write to the Free Software  
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  
02111-1307 USA

Also add information on how to contact you by electronic and  
paper mail.

You should also get your employer (if you work as a programmer)  
or your  
school, if any, to sign a "copyright disclaimer" for the library, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the  
library 'Frob' (a library for tweaking knobs) written by James  
Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!